A report on Project 1: A Checkers-Playing Agent

So, as we increase the search depth is increase, the AI will be much more difficult to defeat. This is because the min max algorithm now travels down even further in the tree to search for a better move with a better score. As it traverses down, it goes through more possible moves that the player might do and simulates it to find the best optimal path for the AI to win. So, we can ultimately say that the deeper the search depth of the min max algorithm goes the more optimal decision the AI makes. However, there is a trade of when doing this. Although by increasing more depth the AI does a more optimal move, it takes too long to come up with a decision because there are too many possibilities moves that the AI is trying to simulate.

Another way we can make the AI more difficult is to improve our score evaluator. The score evaluator give the AI a score to look at whether this or that move will generate more score thus, gaining more control of the board. So, by improving our evaluator, we can not only solve our time problem but also create a better heuristic evaluator function to evaluate different moves. Since we know that time does matter when it comes to the efficiency of a search algorithm, by having a good evaluator we can then help to shorten the time needed to find an optimal path for the AI.